Glossary D

A

A5 The register that points to the application space for the current application.

APDA (Apple Programmer's and Developer's Association) Apple's in-house membership organization that distributes technical information and products to developers.

AppleTalk Apple's networking protocol.

application globals The globals that your application defines.

application parameters Information that the system stores in the application space about a running application.

application space Memory that's available for dynamic allocation by applications.

array window In the THINK Debugger, a window that displays the contents of an array. It is similar to a data window.

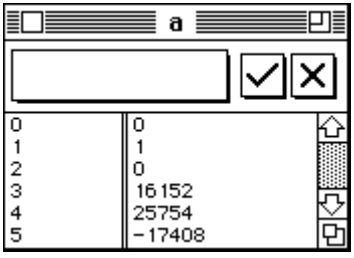

Figure D-1 Array window

auto-indenting Describes an editor that indents lines for you automatically. Whenever you start a new line in the THINK Editor, it's indented by the same amount of space as the line above it.

auto-make The part of the THINK Project Manager that compiles your program for you. Using the information in your project file, it compiles only the files that have been changed since the last compilation.

auto mode The mode in which the THINK Debugger pauses at breakpoints to update the Source and Data windows. To enter auto mode, hold down the Option or Command key as you click any of the status panel buttons (except Stop). To exit auto mode, click the Stop button. See also **breakpoint** and **status panel**.

B

breakpoint A flag you set on a statement telling the debugger to stop execution at that statement. See also **conditional breakpoint**, **simple breakpoint**, and **temporary breakpoint**.

Bundle bit A bit that lets the Finder know that your application has a `'BNDL'` resource. Set the Bundle bit with the **Set Project Type** dialog. The `'BNDL'` resource maps file types to their proper icons.

C

call chain See **subroutine call chain**.

callback routine A routine that you pass as an argument to a Macintosh Toolbox routine.

CODE component One of the four components a project has for each source file or library. Contains the executable code generated for the file. See also **DATA component**, **JUMP component**, and **STRS component**.

code resource A resource that contains a program's code. Applications have code resources of type `'CODE'`, and device drivers and desk accessories have code resources of type `'DRVR'`. Some other types of code resources are `'XCMD'`, `'cdev'`, `'INIT'`, and `'WDEF'`. See also **code segment**.

code segment An individual `CODE` resource, which is part of the code of a Macintosh application. Segments are loaded in and out of memory by the Segment Loader. See also **code resource.**

conditional breakpoint A breakpoint at which the THINK Debugger stops only if a condition is true. See also **breakpoint, simple breakpoint,** and **temporary breakpoint**.

current function In the THINK Debugger, the function that is executing. It's displayed in the lower left corner of the Source window. See also **Source window**.

Figure D-2 Name of the current function

current statement In the THINK Debugger, the statement that the debugger is about to execute.

current statement arrow In the THINK Debugger, the arrow that appears to the left of the current statement.

D

DA Shell A small program that simulates an environment for your desk accessory. Useful for debugging your desk accessories. Included with the THINK development environment.

DATA component One of the four components a project has for each source file or library. Contains the global and static variables for the file. See also **CODE component**, **JUMP component**, and **STRS component**.

data fork One of two forks in any Macintosh file. It contains raw data, such as text or numbers. Compare with **resource fork**.

Data window In the THINK Debugger, the window you use to display and change the values of your variables.

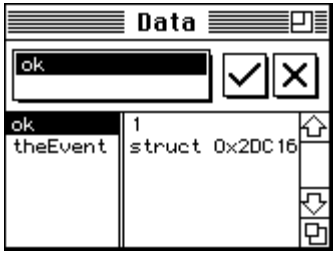

Figure D-3 Data window

Deselect button The button with an "X" in the THINK Debugger's Data window. Click it to erase the contents of the entry field. See also **Data window** and **entry field**.

Figure D-4 Deselect button

desk accessory A small program that is implemented as a device driver and can be used at the same time as an application. Its type is `DFIL`, and its creator is `DMOV`.

Desk Manager The part of the Macintosh Toolbox that handles desk accessories.

device A part of the Macintosh or a piece of external equipment that transfers information into or out of the Macintosh. The most common devices are disk drives, modems, and printers.

device driver A program that controls the exchange of information between an application and a device. See also **device**.

Device Manager The part of the Macintosh Toolbox that supports device input and output.

driver See **device driver**.

E

Enter button The button with a check mark in the THINK Debugger's Data window. Click it to enter the contents of the entry field. See also **Data window** and **entry field**.

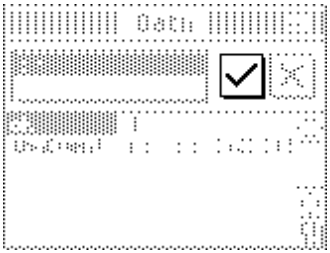

Figure D-5 Enter button

entry field In the THINK Debugger, the field in the upper left corner of the Data window. Here you enter the names of variables you want to view or new values for variables you want to change.

Figure D-6 Entry field

F

Far CODE Code that contains functions that you refer to with 32-bit absolute addresses instead of 16-bit relative addresses. It lets you have a jump table as large as 256K.

Far DATA Code that contains global data that you refer to with 32-bit absolute addresses instead of 16-bit relative addresses. It lets you have an unlimited amount of global data.

File Manager The part of the Macintosh Toolbox that handles file input and output.

G

Grep A powerful pattern-searching feature in the THINK Editor. Based on the `grep` command found in many Unix systems.

H

header file A file that contains function prototypes and variable declarations. It does not contain source code to implement functions. You include it in files that use what it declares.

heap The area of memory in which space is dynamically allocated and released on demand, using the Memory Manager. See also **zone size**.

I

#include (v) To put a file in an `#include` statement. For example, to #include the file `stdio.h` in your source file, put this statement in your source file:

```
#include <stdio.h>
```

(adj) Describes any file you use in an `#include` statement. See also **header file**.

J

JUMP component One of the four components a project has for each source file or library. Contains the jump table for the file. See also **jump table**. Compare with **CODE component**, **DATA component**, and **STRS component**.

jump table A table that contains one entry for every function in an application. The Segment Loader uses this to find a function in one segment when a function in another segment refers to it.

L

library A collection of compiled functions that you can use in many programs.

Link Errors window When you compile a program, this window displays the names of undefined and multiply defined routines.

List Manager The part of the Macintosh Toolbox that helps you create, display, and manipulate lists.

low-memory globals Globals stored at the bottom of the Macintosh memory space. They are available to all running applications, but you should manipulate them only with Toolbox routines.

M

MacHeaders A precompiled header file containing the most common declarations used in Macintosh programs. See also **precompiled header file**.

Macintosh Programmer's Workshop (MPW) A software development environment for the Macintosh made by Apple. A fully configured system contains a shell (with a command-line interface much like Unix), a C compiler, a Pascal compiler, and an assembler.

Macintosh Toolbox The software in the Macintosh ROM that implements the Operating System and the User Interface Toolbox. The Operating System handles low-level operations like I/O and memory management. The User Interface Toolbox helps you implement the standard Macintosh user interface in your application.

MacsBug A low-level debugger from Apple that uses a command-line interface. It can show exactly what's in memory as your program runs, but it can't show you high-level information, like variable names.

marker A placeholder you can put in a source file. You place it with the Mark command, and you go to it by Command-clicking on the title bar of the source window and selecting its name from the pop-up menu.

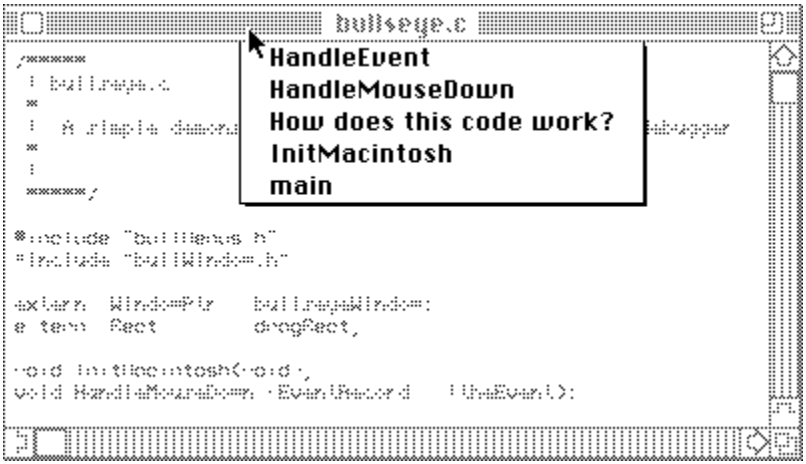

Figure D-7 Marker pop-up menu

MC68000 (1) The microprocessor used in the Macintosh Classic, Plus, Portable, SE, 128K, 512K, and 512K enhanced. (2) A generic term for the microprocessors in the MC68000 family, including the MC68000, MC68020, and MC68030.

MC68020 The microprocessor used in the Macintosh LC and II. In addition to what the MC68000 provides, this provides faster processing, an extended instruction set, and full 32-bit addressing. With the MC68851 coprocessor, this processor also lets you use virtual memory.

MC68030 The microprocessor used in the Macintosh SE/30, IIci, IIcx, IIfx, IIsi, and IIx. In addition to what the MC68020 provides, this processor lets you use virtual memory without the MC68851. See also **virtual memory**.

MC68851 An coprocessor for the Macintosh II that handles paged memory management and lets you use virtual memory. Sometimes called the *Paged Memory Management Unit* or *PMMU*. See also **virtual memory**.

MC68881 A coprocessor that provides high-speed support for processing scientific computations. Sometimes called the *floating-point unit* or *FPU*.

MPW See **Macintosh Programmer's Workshop**.

multi-segment Describes any application, desk accessory, device, or code resource that contains more than one code segment.

O

.o files The binary files produced by MPW compilers. They can be used as libraries in projects.

object code The machine language that a compiler generates from source files. The THINK Project Manager stores it in your project.

optimizer The part of the THINK development environment that makes your code smaller and/or faster.

P

partition size The amount of memory an application gets when running under MultiFinder. You set it in the **Get Info** dialog in the Finder. Also called *Application Memory Size*.

PMMU See **MC68851**.

pointer A variable that contains the address of something in memory, like another variable or a function.

precompiled header file A header file in a format that the THINK Project Manager can use readily, making it faster to load than other headers. Must contain only declarations and preprocessor symbols. See also **header file**.

Print Manager The part of the Macintosh Toolbox that lets programs print on any kind of printer with the same interface.

profiler The part of the THINK development environment that collects timing statistics about your functions. To use it, turn on the Profile option in the Code Generation section of the **Options** dialog.

program Anything you can compile with the THINK Project Manager: an application, desk accessory, device driver, or code resource.

project A file that keeps track of all the files in your program. It contains the object code for your files, maintains the dependencies between them, and remembers what needs to be recompiled. This is the heart of the THINK development environment.

project prefix Text that the THINK Project Manager includes in all the source files in your project. It's as if you put the text at the beginning of all your source files. You set this prefix in the Prefix page of the **Options** dialog.

project tree The folder your project is in and all the folders in it. Keep the files you use in only one project here.

project window The window that displays the files in your project. Here you can select files to edit and debug. See also **project**.

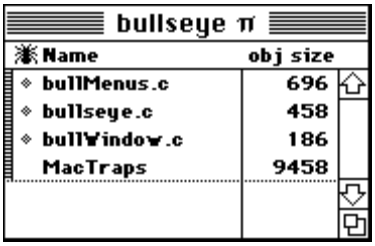

Figure D-8 Project window

Q

QuickDraw The part of the Macintosh Toolbox that performs all the graphic operations on a Macintosh screen.

R

ResEdit An application that lets you interactively create and edit resources. Included with the THINK development environment.

resource A piece of data stored in the resource fork of a Macintosh file. Common resources are windows, menus, fonts, and executable code. See also **resource fork**.

resource file (1) A file that contains the resources your project uses. When you run a program in the environment, the THINK Project Manager opens this file so you can use its resources. When you build your project, the THINK Project Manager copies the

resource file into your finished program. (2) Another name for the **resource fork** of a Macintosh file.

resource fork One of the two forks in any Macintosh file. It contains resources, such as menus, fonts, icons, and executable code. To access it, use the Resource Manager. See also **resource**. Compare with the **data fork**.

S

SANE See **Standard Apple Numerics Environment**.

segment See **code segment**.

Segment Loader The part of the Macintosh Toolbox that loads the code of an application into memory, either as a single unit or divided into dynamically loaded. See also **code segment**.

selected statement In the THINK Debugger, the statement you selected in the source window by clicking on it. It is highlighted. If you don't click on a statement, the current statement is the selected statement. Compare with **current statement**. See also **Source window**.

Serial Driver A device driver that controls communication, via serial ports, between an application and serial devices.

simple breakpoint A breakpoint at which the THINK Debugger always stops. See also **breakpoint**, **conditional breakpoint**, and **temporary breakpoint**.

smart linking A technique the THINK Project Manager uses to remove as much unused code from your final program as possible. The THINK Project Manager removes code for all unused functions.

source files The text files that contain your source code.

Source window In the THINK Debugger, the window that contains the source code being executed and the status panel.

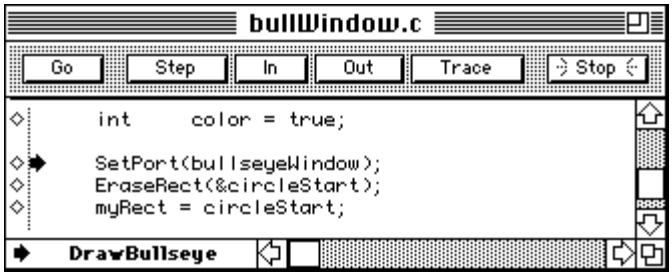

Figure D-9 Source window

stack (1) Any area of memory that is allocated and deallocated in a last-in first-out order, like a stack of trays in a cafeteria. (2) A specific stack in the MC68000 that contains information on each routine in the subroutine call chain. Each time you call a routine, a stack frame is added that contains the routine's parameters, variables, and return address. When you return from a routine, its frame is removed from the stack.

stack frame A frame in the MC68000 stack that contains a routine's parameters variables, and return address.

stack size The maximum size of the stack in the MC68000. By default, this is 16K.

Standard Apple Numerics Environment (SANE) A system for performing floating-point arithmetic that Apple computers use. It meets all requirements for extended-precision, floating-point arithmetic as prescribed by IEEE Standard 754 and ensures that all floating-point operations are performed consistently and return the most accurate results possible.

statement marker In the THINK Debugger, the diamond to the left of a line of source code. It shows that the line generates code.

status panel In the THINK Debugger, the top of the source window. It contains six buttons that let you control the execution of your program.

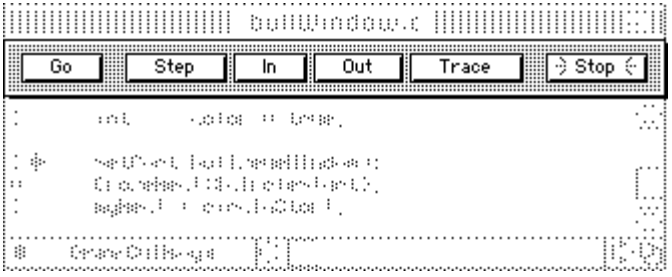

Figure D-10 Status panel

string literal A sequence of characters enclosed by quotes.

STRS component One of the four components a project has for each source file or library. If you use the Separate STRS option, this contains the string literals and floating-point constants for the file. Otherwise, they're in the DATA component and this component doesn't exist. See also **CODE component**, **DATA component**, and **JUMP component**.

struct window In the THINK Debugger, a window that displays the contents of a `struct`. It is similar to a data window.

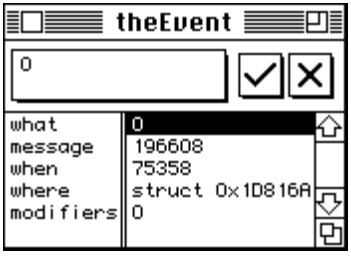

Figure D-11 Struct window

subroutine call chain The list of functions called to get to the current function. In the THINK Debugger, click on the current function indicator to see this. See also **current function**.

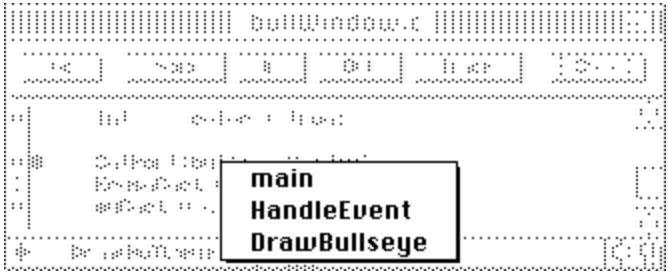

Figure D-12 Subroutine call chain

T

temporary breakpoint Lets you set a breakpoint and start execution in one step. The next time the THINK Debugger comes to it, the debugger stops and clears it. See also **breakpoint**, **conditional breakpoint**, and **simple breakpoint**.

THINK tree The THINK Project Manager folder and all the folders in it. If you use a file in more than one project, keep it in here.

TMON A low-level debugger that uses a graphical interface. It can show exactly what's in memory as your program runs, but it can't show high-level information, like variable names.

Toolbox See **Macintosh Toolbox**.

Toolbox routine Any routine in the Macintosh ROM.

trap intercept routine A routine that executes instead of a trap routine that implements a Toolbox routine.

V

vertical retrace task A task that executes during the vertical retrace interval, which occurs sixty times a second.

virtual memory Space on your hard disk that you can use as though it were extra RAM.

Z

zone See **heap**.

zone size The size of your application's heap.

D Glossary